



Quarkus Testing

JEE Microservices

@ CGS IT – 2023

Version 1.0.5

Inhalt

- Was ist Quarkus?
- Quarkus - Architektur
- GraalVM
- Quarkus Vorteile
- Quarkus Container CDI
- Quarkus – Spring DI Compatibility
- Quarkus CDI Limitations
- Quarkus Installation Requirements
- Quarkus Installation -

Quarkus Tests

- Simple Container based Testing Unit + CDI + Quarkus
- Component Tests via `@QuarkusTest`
- Integration tests via `@QuarkusIntegrationTest`
- Einfaches Testen von CDI Beans, Rest und DB Objekten
- Unterstützung von RestAssured für Rest API Tests

Quarkus Testing - Grundlagen

- Basis Junit 5
- Rest Assured Library für einfaches api testen

```
<dependency>
  <groupId>io.quarkus</groupId>
  <artifactId>quarkus-junit5</artifactId>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>io.rest-assured</groupId>
  <artifactId>rest-assured</artifactId>
  <scope>test</scope>
</dependency>
```

Test DB Klasse

- Inject des Repositories direkt in den Test möglich
- Initialisierung der DB und des gesamten JPA via CDI Quarkus Test Containers

```
@QuarkusTest
class ChatMessageRepositoryTest {

    @Inject
    Logger log;

    @Inject
    ChatMessageRepository cmRepository;

    @Test
    void readChatMessage() {
        ChatMessageEntity cmEntity =
        cmRepository.readChatMessage(1L);
        assertNotNull(cmEntity);
        log.info("cmresult" + cmEntity.toString());
    }
}
```

Quarkus Test - Example

- `@QuarkusTest` initialisiert den gesamten Quarkus Server soweit benötigt
- `@TestHTTPEndpoint` Annotation setzt alle test roots auf die Base URL der `ExampleResource` `/hello`

```
@QuarkusTest
@TestHTTPEndpoint(ExampleResource.class)
public class ExampleResourceTest {

    @Test
    public void testHelloEndpoint() {
        given()
            .when()
            //.get("/hello")
            .get()
            .then()
            .statusCode(200)
            .body(is(notNullValue()));
            //.body(is("Hello RESTEasy"));
    }
}
```

Rest Assured Grundlagen

```
Given().  
    param("x", "y").  
    header("z", "w").  
when().  
Method().  
Then().  
    statusCode(XXX).  
    body("x", "y", equalTo("z"));
```

Code	Explanation
Given()	'Given' keyword, lets you set a background, here, you pass the request headers, query and path param, body, cookies. This is optional if these items are not needed in the request
When()	'when' keyword marks the premise of your scenario. For example, 'when' you get/post/put something, do something else.
Method()	Substitute this with any of the CRUD operations(get/post/put/delete)
Then()	Your assert and matcher conditions go here

Quarkus – Rest – Testing Objects

- Extract Response Body und Deserialisierung in DTO zur Validierung

```
@QuarkusTest
class ChatMessageResourceTest {

    @Test
    @Disabled
    public void readChatMessage() {

        ResponseBody body =
            given()
            .when().get("/chatMessageResource/chatMessage/10")
            .then().extract().response().body();

        ChatMessageDTO dto = body.as(ChatMessageDTO.class);
        assertNotNull(dto);
    }
}
```


Rest Assured

– POST Object

```
@Test
@Disabled
public void createChatMessagePost() {

    ChatMessageDTO chatMessageDto = new ChatMessageDTO();

    RequestSpecification httpRequest = given() // .header("Authorization",
"Bearer " + token)
        .header("Content-Type", "application/json");

    RequestSpecification rs = httpRequest.body(chatMessageDto);

    Response response = rs.post("/chatMessageResource/chatMessage");

    //Fetching the response code from the request and validating the same
    System.out.println("The response code - " + response.getStatusCode());
    assertEquals(response.getStatusCode(), 200);
}
}
```

```
RuntimeException thrown = Assertions.assertThrows(RuntimeException.class, () -> {  
    List<ChatMessageEntity> cmEntity =  
    cmRepository.findChatMessagesWithLikeNameAndOrdedByDateWithQueryBuilder("room%");  
});  
Assertions.assertNotNull(thrown);
```

Danke für Ihre Aufmerksamkeit